REMARKS

Claims 17-36 are pending in the application. Claims 17-36 stand rejected. Applicants herein amend claim 17, 21, 24, 25, 26-30, and 32. Further review and consideration is respectfully requested.

Claim Rejections – 35 USC § 103

Claims 17-27 and 29-34 stand rejected under 35 U.S.C. § 103(a) over U.S. Patent No. 6,343,287 to Kumar in view of U.S. Patent Application Publication No. 2004/0078568 to Pham.

According to section 2141 of the MPEP, the Office may reject a claim as obvious in the instance that "the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains." MPEP, § 2141. Rejections based on obviousness cannot be sustained by conclusory statements. Rather, such conclusion must be supported "evidence which as a whole shows that the legal determination sought to be proved (i.e., the reference teachings establish a *prima facie* case of obviousness) is more probable than not." MPEP, § 2142. (Emphasis original)

Turning to claim 1, the Office asserts:

> However, Pham et al. teaches wherein the operating system includes a database management program that encapsulates a file system and the operating system is configured to store, data in the file system as file streams, and generate, items associated with the file streams in the database management program (See Paragraphs 0032, 0036-0037 and 0049, and figures 1-2).

Action, p. 3. Applicants have amended claim 17 to clarify and respectfully submits that Kumar in view of Pham fail to teach or suggest at least:

> the operating system including a database management program that encapsulates a file system, the database management program *encapsulating the file system by exclusively handling file system*

> *access requests from a group of user mode applications*, the group of user mode applications being configured to interface with an operating system application program interface, the operating system application program interface being configured to send read/write requests issued by the group of user mode applications to the database management program of the operating system, the database management program of the operating system being configured to open files stored in the file system in response to receipt of read/write requests.

Support for this subject matter can be found in at least paragraphs [1057] through [1068] and [1007] – [1021]. Applicants appreciate that some of this subject matter has been newly added and has not been examined. However, since this newly added subject matter merely clarifies how the database management program encapsulates the file system Applicants will address the applicability of Kumar and Pham to it in order to expedite examination.

Regarding FIG. 1 of Pham, Applicants respectfully submit that it fails to teach the aforementioned subject matter. In particular, FIG. 1 fails to illustrate a database management program that encapsulates a file system. Instead, it seems to show a security interposer sitting between the operating system and file system:

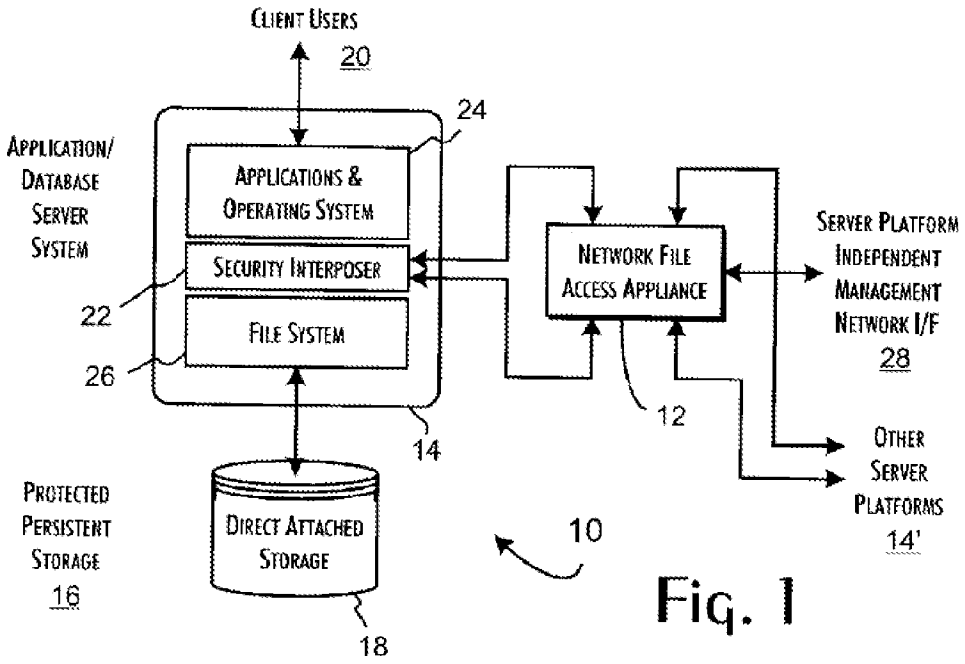

Fig. 1

Nothing in the figure, on its face, indicates that a database encapsulates the file system. Rather, FIG. 1 of Pham shows a server including applications 24, a security interposer 22, a file system 26, and a network file access appliance 12. The functionality of security interposer 22 is described in paragraph [0033] as follows:

> [**0033**] The security interposer layer **22** selectively routes file oriented data transfers between the operating system kernel **24** and file system **26** through the secure network file access appliance **12** to encrypt and decrypt the file data stored to the protected persistent storage resources **16** subject to access policies implemented within the secure network file access appliance **12**. In accordance with the present invention, the file data encryption maintains the logical file-oriented structure of the data and is thus transparent to the persistent storage resources **16**. Additionally, the secure network file access appliance **12** can implement IP firewall functions, limiting potential attacks on the security of the secure network file system appliance from the computer server platforms **14**, **14'**.

Instead of determining what file corresponds to an item and de-serializing the file into an item, the security interposer layer 22 seems to encrypt and decrypt data as it flows through the security interposer layer 22. Simply put, Pham describes security interposer layer 22 as an encryption filter instead of a database management program.

Like FIG. 1, paragraph [0032] similarly fails to describe a database management program that encapsulates a file system. Instead, it generally describes the elements in FIG. 1 and does not indicate that a database management program encapsulates a file system. It reads as follows:

> [0032] The present invention utilizes and extends the operation of a secure network file system appliance to establish a security envelope protecting persistent stored data accessible through various computer system platforms. An exemplary server platform protected environment **10** is shown in **FIG. 1**. A secure network file access appliance **12** is preferably implemented in the environment **10** to support the secure operation of one or more computer server platforms **14, 14'** relative to protected persistent storage resources **16,** such as direct attached storage **18**. For purposes of the preferred embodiments of the present invention, the computer system platforms **14, 14'** are database and application server platforms supporting local and remote client systems and users **20**. The secure network file system appliance is integrated with the computer system platform **14** through a security interposer layer **22** established between the operating system kernel **24** and a file system **26** through which data is transferred relative to direct attached storage **18**.

Notably absent from the text of paragraph [0032] is any indication that a database management program is used. Even more to the point, nothing in the text indicates that user mode applications exclusively access a file system through a database management program.

Reliance on FIG. 2 as illustrating the claimed subject matter is similarly misplaced. According to Pham, FIG. 2 shows a detailed architectural block diagram of a preferred embodiment.

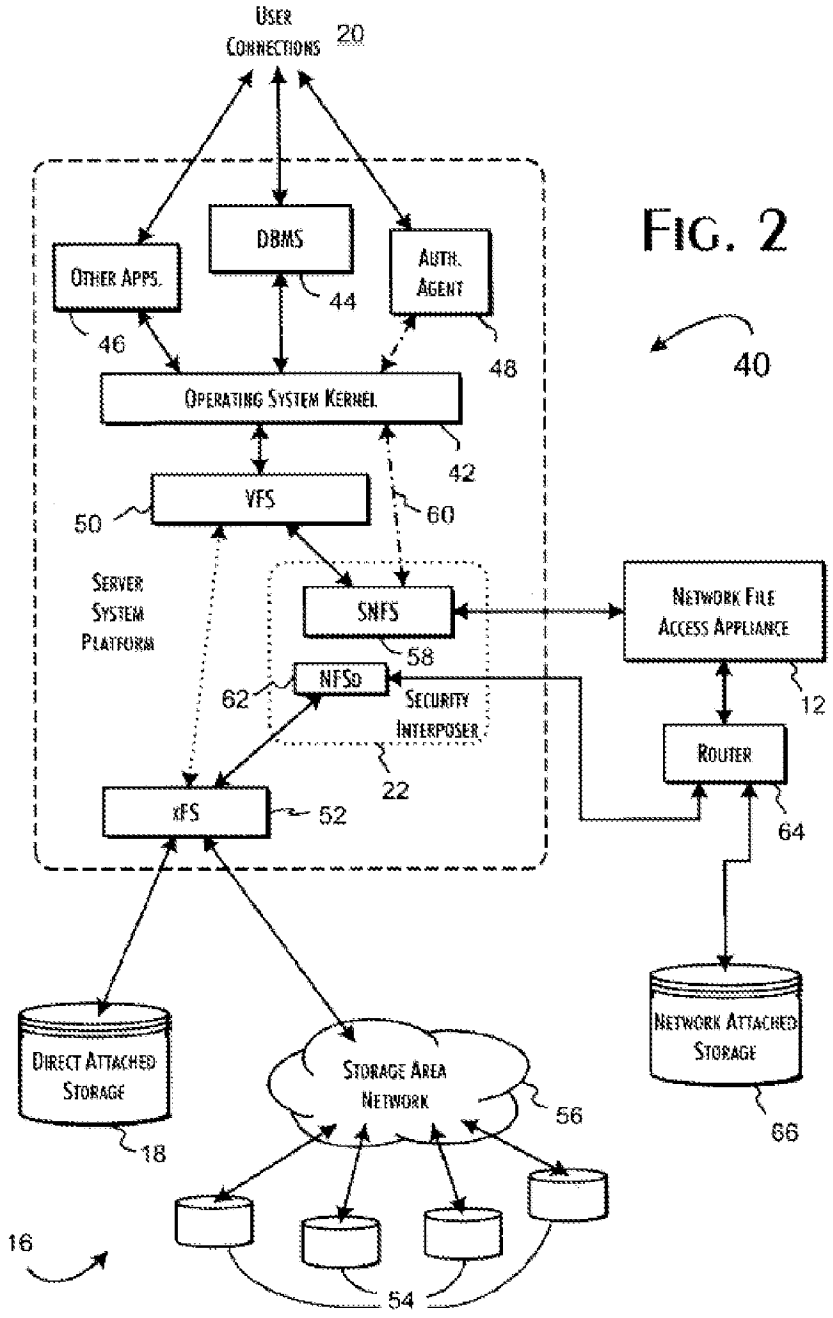

FIG. 2

FIG. 2 clearly shows a database management 44 as an element that is separate from the operating system kernel 42. In addition, the figure shows applications interacting with the virtual file system 50 through the operating system kernel 42. As such, FIG. 2 seems to show applications

directly accessing the kernel instead of accessing the file system through the database. This architecture is completely different than the claimed subject matter.

The text of paragraph [0036] also undercuts the Office's assertion. Rather than describing an architecture where a database management program runs in the kernel and applications access the file system through it, the paragraph indicates that applications and the database management program access the file system the same the same way. The paragraph reads as follows:

> **[0036]** The preferred structure **40** of a computer system platform **14** is shown in **FIG. 2**. The platform **14** conventionally includes an operating system kernel **42** supporting execution of applications, such as a database management system (DBMS) **44** and other server applications **46**, in a user mode execution space. The operating system kernel **42** also preferably supports execution of an authentication agent program **48** substantially, if not completely, within a kernel mode execution space. A virtual file system switch (VFS) **50** provides a conventional interface to any number of different conventional file systems (xFS) **52** as necessary to access conventional direct attached storage **18** and, for example, storage devices **54** accessible through a storage area network **56**.

Contrary to the Office's assertions, paragraph [0036] indicates that the user mode applications 46 do not access the file system through the database management system 44.

Paragraph [0037] similarly fails to describe the claimed subject matter. Instead, it seems to describe the security interposer layer 22 as a file system. According to Pham, the security interposer layer 22 acts as a gatekeeper in that it determines whether or not to allow an access request to proceed. In particular, paragraph [0037] reads as follows:

> [0037] In a first preferred embodiment of the present invention, the security interposer layer **22** is implemented utilizing a secure network file system (SNFS) **58** and a conventional network file system (NFS) client file system daemon (NFSd) **62**. The secure network file system **58** is preferably based on a conventional network file system (NFS) implementation used to route network file transfer requests and data through the secure network file access appliance **12**. The secure network file system **58** includes modifications to enable collection of user, process and session information through an interface **60** to the operating system kernel **42**, in regard to specific network file transfer requests, and to provide this information to the secure network file access appliance **12** as a basis for determining whether to permit the corresponding network file transfer to proceed.

Here, it seems that requests are merely routed through the security interposer layer 22. In contrast, the database management program of claim 1 identifies data in the file system in response to queries and provides it to user mode applications as objects.

Similar to paragraph [0037], paragraph [0049] fails to teach or suggest the claimed subject matter. As far as Applicants can discern, paragraph [0049] describes the a virtual file system 132 and like paragraph [0037] it is missing a database management system according to claim 1:

> [0049] **FIG. 5A** shows a configuration **130** of the computer system platform **14** employing the second preferred embodiment of the security interposer layer **22** where the secure virtual file system **132** effectively combines the function of the secure network file system **58** and network file system daemon **62**. While the secure virtual file system **132** may also use the NFS protocol for transferring file transfer requests and data with the network file access appliance **12**, the secure virtual file system **132** preferably implements a non-standards compliant RPC-based message

> transfer protocol to obscure the information transferred between the computer system platform **14** and network file access appliance **12**. A conventional interface to the virtual file system switch **50** is supported so that the secure virtual file system **132** appears to the switch **50** as an ordinary file system. The secure virtual file system **132** implements the extended operating system kernel **42** interface **60** to support operation of the authentication agent program **48**. The secure virtual file system **132** also implements a conventional file system overlay interface **134**, permitting functional capture and utilization of conventional file systems **52**. Dedicated or proprietary file systems **136** may also be closely coupled to the secure virtual file system **132**.

The text of paragraph [0049] seems to describe the virtual file system 132 as including the functionality of the secure network file system 58 and network file system daemon 62. The text clearly describe a protocol used to transfer information between computer system platform 14 and network file access appliance 12 and interfaces to the operating system kernel 42 as well as the virtual file system. Notably absent from this paragraph is any discussion of the operating system as including a database management program that encapsulates a file system. In fact, the phrase "database management program" is completely absent from this paragraph.

The forgoing analysis shows that the sections of Pham cited as teaching the aforementioned feature of claim 17 fail to teach what is claimed. As such, Applicants respectfully submit that the combination of Kumar and Pham fail to render claim 17 obvious. Accordingly, Applicants respectfully request reconsideration of the rejection of claim 17.

Insomuch as claims 18-27 and 29-34 depend from claim 17, Applicants submit that the combination of Kumar and Pham fail to render these claims obvious for at least the reasons set forth above with respect to claim 17. Accordingly, Applicants respectfully request reconsideration of the rejections of claims 18-27 and 29-34.

Claim 28 stands rejected under 35 U.S.C. § 103(a) over Kumar in view of Pham and U.S. Patent No. 6,473,851 to Plutowski.

Insomuch as claim 28 depends from claim 17, Applicants submit that the combination of Kumar, Pham, and Plutowski fail to render this claim obvious for at least the reasons set forth

above with respect to claim 17. Accordingly, Applicants respectfully request reconsideration of the rejection of claim 28.

Claims 35 and 36 stand rejected under 35 U.S.C. § 103(a) over Kumar in view of Pham and C. Liebig et al "A Publish/Subscribe COBRA Persistent State Service Prototype. Middleware 2000: IFIP/ACM International Conference of Distributed Systems Platforms," New York, NY, USA, April 2000.

Insomuch as claims 35 and 36 depend from claim 17, Applicants submit that the combination of Kumar, Pham, and Liebig fail to render this claim obvious for at least the reasons set forth above with respect to claim 17. Accordingly, Applicants respectfully request reconsideration of the rejection of claims 35 and 36.

CONCLUSION

Applicants request the Examiner reconsider the rejections and issue a Notice of Allowance of all the claims.

Date: May 2, 2011

/David M. Platz/
David M. Platz
Registration No. 60,013

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439